Building Interoperable Healthcare Apps with Google Cloud FHIR APIs

Dharmesh Patel
Google Cloud Healthcare & Life Sciences
Agenda

1. Healthcare Interoperability & FHIR
2. Google’s Cloud Healthcare FHIR API
3. SMART on FHIR
Healthcare Interoperability & FHIR
The Interoperability Challenge in Healthcare
FHIRM as the Data Model and API Spec for Interoperability

FHIRM® – Fast Healthcare Interoperability Resources (FHIRM, pronounced "fire") is a next generation healthcare data standard created by HL7 to describe data formats and elements (known as "resources") and application programming interfaces (APIs) for exchanging healthcare data (clinical, diagnostic, medications, workflows, financial, etc)

- F – Fast (to design & to implement)
- H – Healthcare
- I – Interoperability
- R – Resources

Google Cloud
Patient Schema in FHIR

FHIR Id & Metadata

Human Readable Summary

Extension with reference to its definition

Standard Data Content:

- Patient Identity
- Name
- Gender
- Date of Birth
- Provider

```
{  
  "resourceType": "Patient",
  "id": "example",
  "meta": {  
      "versionId": "1",
      "lastUpdated": "2017-01-03T16:05:00.792Z"
  },
  "text": {  
      "status": "generated",
      "div": "<div xmlns="http://www.w3.org/1999/xhtml"><p>Henry Levin the 7th</p></div>"
  },
  "extension": [  
      {  
          "url": "http://hl7.org/fhir/StructureDefinition/us-core-birthsex",
          "valueCode": "M"
      }
  ],
  "identifier": [  
      {  
          "use": "usual",
          "system": "urn:oid:1.2.36.146.595.217.0.1",
          "value": "12345"
      }
  ],
  "active": true,
  "name": [  
      {  
          "use": "official",
          "family": "Levin",
          "given": [ "Henry" ],
          "suffix": [ "the 7th" ]
      }
  ],
  "gender": "male",
  "birthDate": "1974-12-25",
  "managingOrganization": {  
      "reference": "Organization/example"
  }
}
```
FHIR for Healthcare App Developers

- Leverage the standard data model instead of building your own custom schemas as you go.

- Use standard FHIR APIs for data access, search, and common operations:
  - Create, Read, Update, Delete
  - History, Search, Validate, Patient record & Transaction

- Easily consume data from other systems:
  - Electronic Health Record (EHR) Systems
  - Health Plans
  - Other healthcare apps
Now, let’s see Google FHIR APIs in Action...
Cloud Healthcare API - Overview

- Serverless/managed service for storing and managing healthcare data privately and securely in the Cloud
- Supports compliance with leading healthcare focused regulatory frameworks such as HIPAA, HITRUST and GDPR.

Data Standards:
- HL7® FHIR Resources and REST APIs
- HL7v2 Messages and REST APIs
- DICOM Instances and REST APIs

Integrated Capabilities:
- Consent Management
- Healthcare De-Identification
Interactions with FHIR Store

Interact
- Create
- Get
- List
- Delete
- Search
- Subscribe

Import
- Bulk

De Identify
- FHIR Config

Notify
- Pub/Sub

Export
- Streaming
- Bulk

Google Cloud
Storing and Accessing FHIR data
Organization: Project -> Location -> Dataset -> FHIR Store
# Accessing FHIR resources using REST

```plaintext
https://healthcare.googleapis.com/<v>/projects/<P>/locations/<L>/datasets/<D>/<type>Stores/<S>
```

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;V&gt;</td>
<td>Healthcare API version</td>
<td>v1</td>
</tr>
<tr>
<td>&lt;P&gt;</td>
<td>Project identifier</td>
<td>ha-portal</td>
</tr>
<tr>
<td>&lt;L&gt;</td>
<td>Location identifier</td>
<td>us-central1</td>
</tr>
<tr>
<td>&lt;D&gt;</td>
<td>Dataset identifier</td>
<td>hc_ds</td>
</tr>
<tr>
<td>&lt;type&gt;</td>
<td>Data type slug</td>
<td>hl7v2, dicom, fhir</td>
</tr>
<tr>
<td>&lt;S&gt;</td>
<td>Store identifier</td>
<td>hc_fs</td>
</tr>
</tbody>
</table>
A-1. Create a Dataset

```
curl -X POST \
  --data "" \n  -H "Authorization: Bearer $(gcloud auth print-access-token)" \n  -H "Content-Type: application/json; charset=utf-8" \n  "https://healthcare.googleapis.com/v1/projects/ha-portal/locations/us-central1/datasets?datasetId=hc_ds"
```
A-2. Create a FHIR Data Store

Can also edit, delete, list stores from UI or API method calls

```
curl -X POST \
   --data '{
      'name': 'projects/ha-portal/locations/us-central1/datasets/hc_ds/fhirStores/hc_fs',
      'enableUpdateCreate': true,
      'version': 'R4'
   }
   -H "Authorization: Bearer $(gcloud auth print-access-token)"
   -H "Content-Type: application/json; charset=utf-8" \
```
curl -X POST \
    --data "{
      'contentStructure': 'RESOURCE',
      'gcsSource': {
        'uri': 'gs://interop-test-data/fhir_r4/ndjson/Patient.ndjson'
      }
    }" \
    -H "Content-Type: application/json; charset=utf-8" \ 
    -H "Authorization: Bearer $(gcloud auth print-access-token)"

C-1. Create a FHIR Resource

```
curl -X POST \
  --data @patient.json \
  -H "Content-Type: application/json; charset=utf-8" \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
```
C-2. Accessing FHIR data: Get Patient

```
curl -X GET "https://healthcare.googleapis.com/v1/projects/ha-portal/locations/us-central1/datasets/hc_ds/fhirStores/hc_fs/fhir/Patient/eedd9961-c6c0-46b4-b397-537b949f4833"
-H "Authorization: Bearer $(gcloud auth print-access-token)"
```
C-3. Updating FHIR data: PUT Patient

```
curl -X PUT \
   --data @patientWithID.json \n   -H "Content-Type: application/json; charset=utf-8" \n   -H "Authorization: Bearer $(gcloud auth print-access-token)"

"https://healthcare.googleapis.com/v1/projects/ha-portal/locations/us-central1/datasets/hc_ds/fhirStores/hc_fs/fhir/Patient/eedd9961-c6c0-46b4-b397-537b949f4833"
```
C-4. Updating FHIR data: PATCH Patient

```bash
curl -X PATCH \
   -H "Authorization: Bearer $(gcloud auth print-access-token)" \
   -H "Content-Type: application/json-patch+json" \
   --data '[
      { "op": "replace", "path": "/birthDate", "value": "2007-09-07" }
    ]'

"https://healthcare.googleapis.com/v1/projects/ha-portal/locations/us-central1/datasets/hc_ds/fhirStores/hc_fs/fhir/Patient/eedd9961-c6c0-46b4-b397-537b949f4833"
```
C-5. Accessing FHIR data: Search Patient

```bash
curl -X GET \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
```
C-6. Accessing FHIR data: Delete Patient

```
curl -X DELETE \n  -H "Authorization: Bearer $(gcloud auth print-access-token)"
  "https://healthcare.googleapis.com/v1/projects/ha-portal/locations/us-central1/datasets/hc_ds/fhirStores/hc_fs/fhir/Patient/eedd9961-c6c0-46b4-b397-537b949f4833"
```
Getting Started with FHIR APIs

1. Introduction to the Cloud Healthcare API: https://cloud.google.com/healthcare

2. Authenticating to the Cloud Healthcare API: https://cloud.google.com/healthcare/docs/how-tos/authentica tion


REST Resource:
projects.locations.datasets.fhirStores.fhir

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation-lastn</td>
<td>Retrieves the N most recent Observation resources for a subject matching search criteria specified as query parameters, grouped by Observation.code, sorted from most recent to oldest.</td>
</tr>
<tr>
<td>Patient-everything</td>
<td>On success, the response body will contain a JSON-encoded representation of a Bundle resource of type searchset, containing the results of the operation.</td>
</tr>
<tr>
<td>Resource-purge</td>
<td>Deletes all the historical versions of a resource (excluding the current version) from the FHIR store.</td>
</tr>
<tr>
<td>capabilities</td>
<td>Gets the FHIR capability statement (STU3, R4), or the conformance statement in the DSTU2 case for the store, which contains a description of functionality supported by the server.</td>
</tr>
<tr>
<td>conditionalDelete</td>
<td>Deletes FHIR resources that match a search query.</td>
</tr>
<tr>
<td>conditionalPatch</td>
<td>If a resource is found based on the search criteria specified in the query parameters, updates part of that resource by applying the operations specified in a JSON Patch document.</td>
</tr>
<tr>
<td>conditionalUpdate</td>
<td>If a resource is found based on the search criteria specified in the query parameters, updates the entire contents of that resource.</td>
</tr>
<tr>
<td>create</td>
<td>Creates a FHIR resource.</td>
</tr>
<tr>
<td>delete</td>
<td>Deletes a FHIR resource.</td>
</tr>
<tr>
<td>executeBundle</td>
<td>Executes all the requests in the given Bundle.</td>
</tr>
<tr>
<td>history</td>
<td>Lists all the versions of a resource (including the current version and deleted versions) from the FHIR store.</td>
</tr>
<tr>
<td>patch</td>
<td>Updates part of an existing resource by applying the operations specified in a JSON Patch document.</td>
</tr>
<tr>
<td>read</td>
<td>Gets the contents of a FHIR resource.</td>
</tr>
<tr>
<td>search</td>
<td>Searches for resources in the given FHIR store according to criteria specified as query parameters.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the entire contents of a resource.</td>
</tr>
<tr>
<td>vread</td>
<td>Gets the contents of a version (current or historical) of a FHIR resource by version ID.</td>
</tr>
</tbody>
</table>
Introduction to SMART

SMART®

Google Cloud
SMART

Substitutable Medical Applications and Reusable Technologies

- Open standards to integrate third-party Apps with EHRs/Patient Portal
  - OpenID Connect and OAuth2 based
- Reusable Apps
  - Write once run with any EHRs!
  - Run in different contexts (EHRs and Patient Portals)
- Secure and seamless access to EHR data in Apps
SMART standardizes healthcare applications

Substitutable Medical Applications and Reusable Technologies

Healthcare Applications

SMART

UX Integration
Single Sign On
Authorization
Clinical Data and Context

Clinical Systems

EHRs
Patient Portals
Data Warehouses
SMART Standalone launch

1. Launch an App

1a. Data access requested (scopes)

2. Authorize Access

2a. Access token for data access, user identity, context (current patient, encounter), UI related information

3. Display Data

3a. FHIR API request with OAuth2 Access token

3b. FHIR resources

FHIR Server

Healthcare APIs / EHR

User App
SMART EHR launch

1. Launch an App from the registered Apps
2. Authorize Access
3. Display Data

- EHR App
  1a. Launch information (server URL, token)
  1b. Data access requested (scopes)
  2a. Access token for data access, user identity, context (current patient, encounter), UI related information
  3a. FHIR API request with OAuth2 Access token
  3b. FHIR resources

- EHR
  1. Launch an App from the registered Apps

- FHIR Server
1a. Data access requested (JWT Assertion)

2a. Access token for data access with authorize system scopes

3a. FHIR API request with OAuth2 Access token

3b. FHIR resources

SMART Backend Services Authorization
SMART Authorization Scopes

- Scope conveys what access an app needs

\[ \text{patient/Immunization.read} \]
- Access type: \text{read}
- FHIR Resource: \text{Immunization}
- Permission: \text{patient}
- App can read Immunization information of a patient in the context

Examples:

- \text{patient/Patient.read} - App can access demographics information of a patient in the context
- \text{patient/*.read} - App can read all resources of a patient in the context
- \text{user/Patient.write} - App can write (create/update/delete) all Patient resources accessible to user
- \text{user/*.write} - App can read all resources of all patients accessible to user
- \text{system/Patient.write} - App can write (create/update/delete) all Patient resources
- \text{system/*.read} - App can read all resources of all patients
SMART Sandbox and App Launcher

http://launch.smarthealthit.org/
Thank you